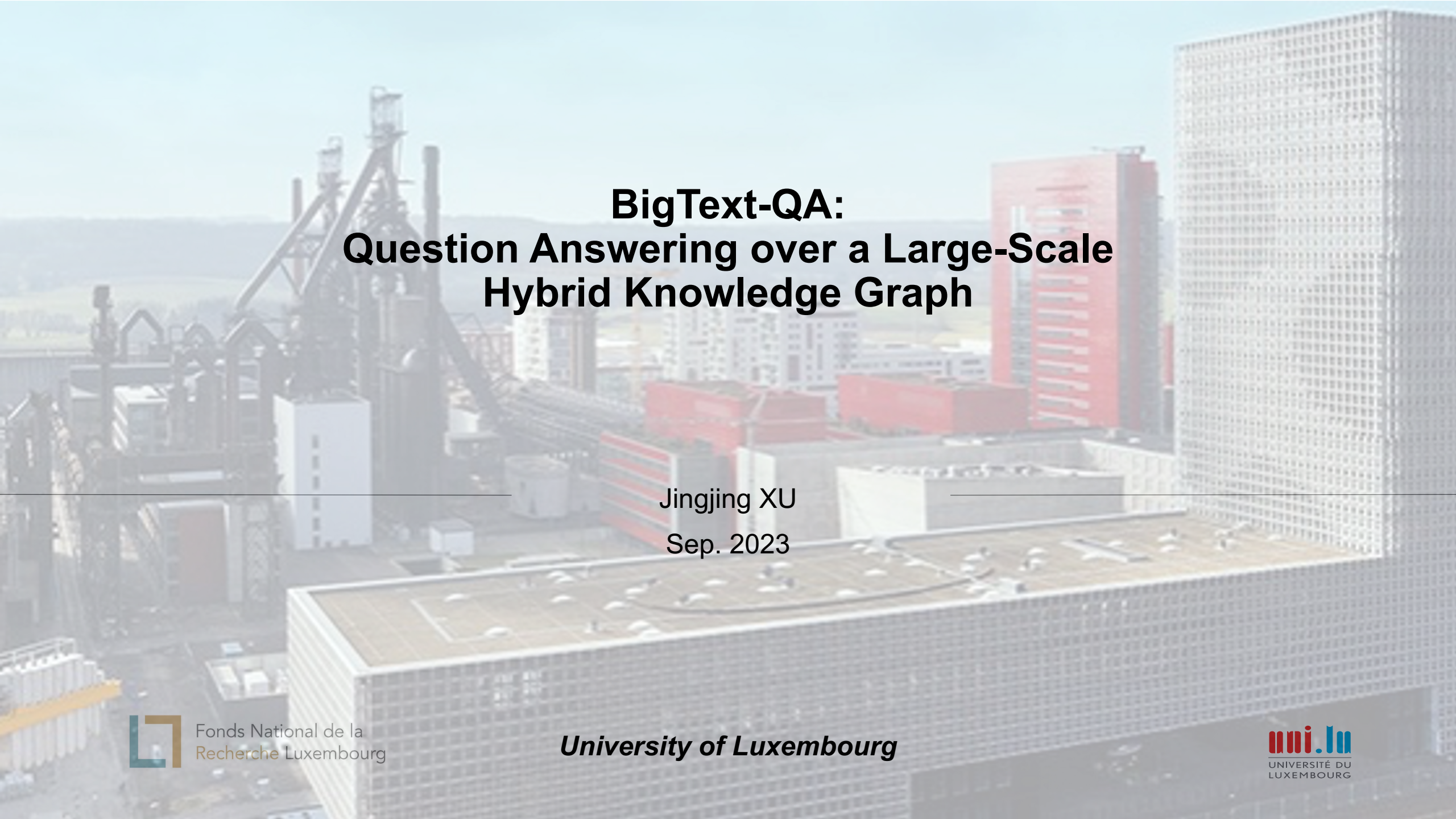


BigText-QA: Question Answering over a Large-Scale Hybrid Knowledge Graph

Jingjing XU

Sep. 2023

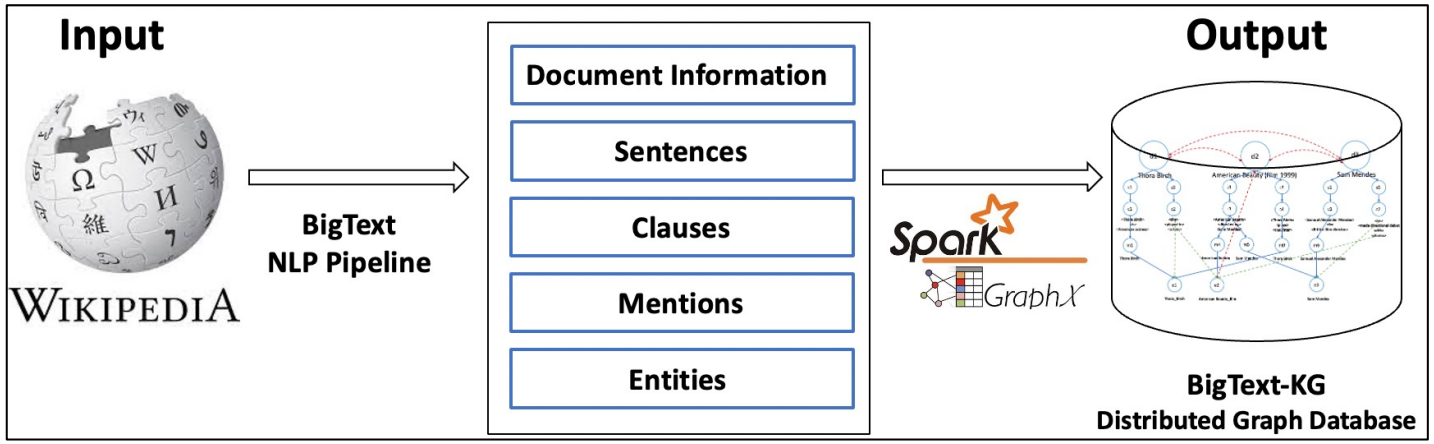
Fonds National de la Recherche Luxembourg

University of Luxembourg

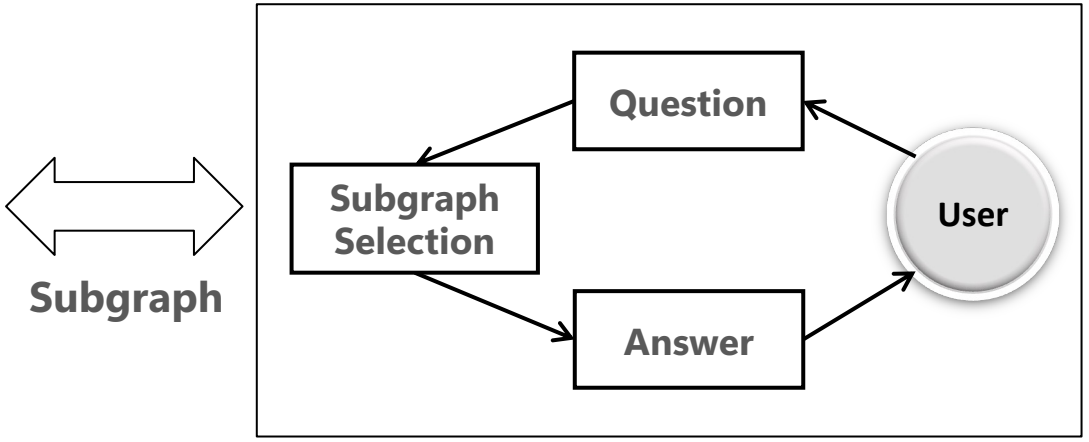
uni.lu UNIVERSITÉ DU LUXEMBOURG

Content

- Motivation
- **BigText-KG:** Distributed **K**nowledge **G**raph Database
- **BigText-QA: Q**uestion **A**nswering System
- BigText-QA Results



Building **BigText-KG** Pipeline

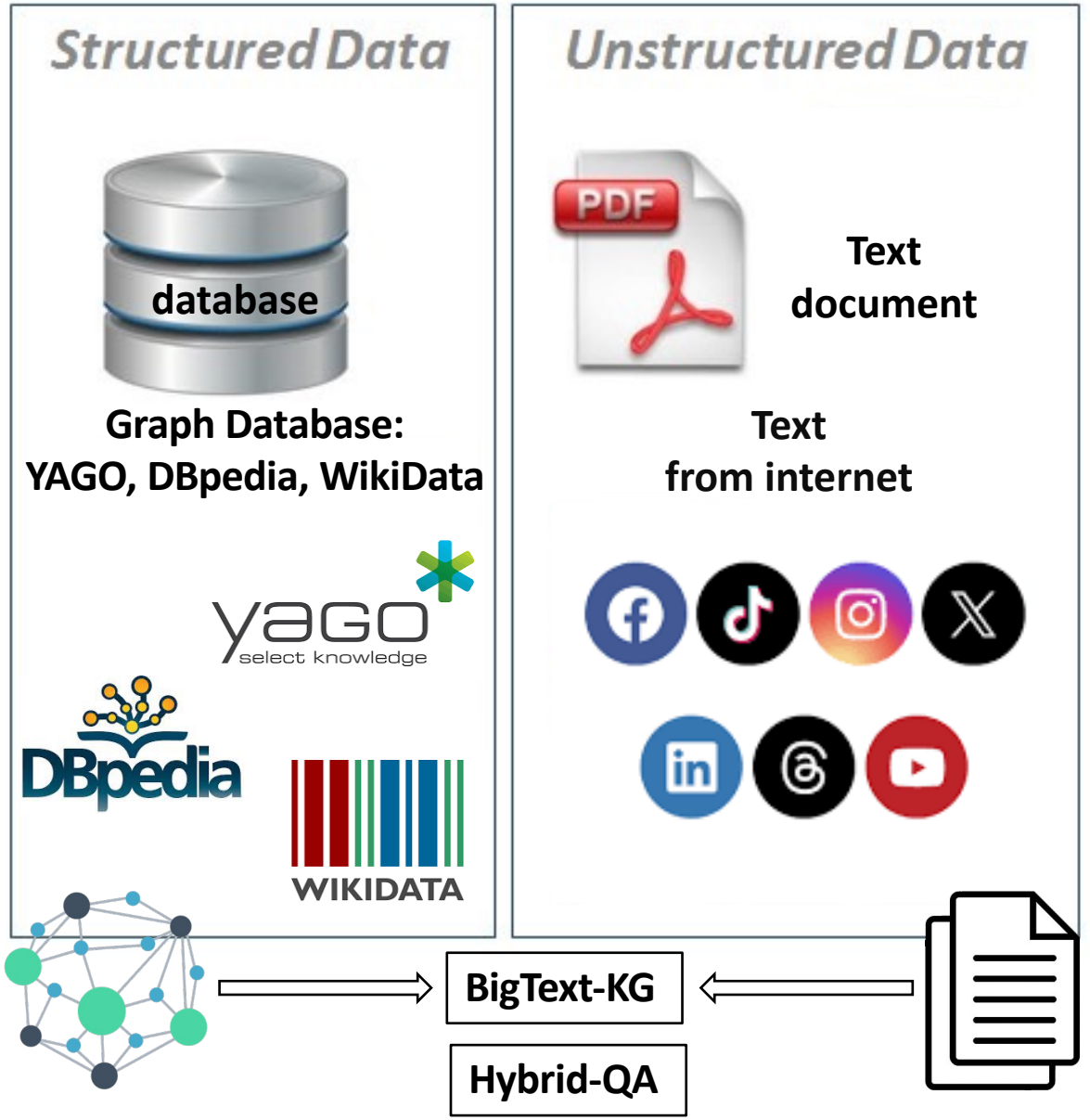


Building **BigText-QA** Pipeline

Motivation

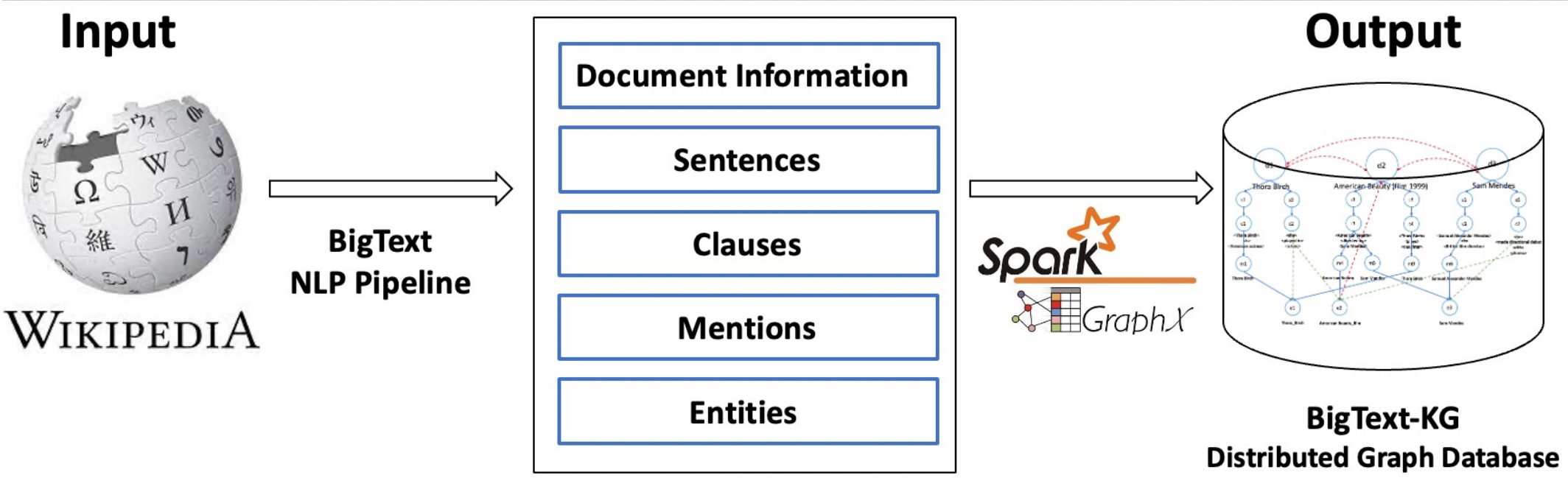
- Current QA approaches (based on input data)
  - KG-based QA (structured input data)
  - Text-based QA (unstructured input data)
  - Hybrid-QA (structured + unstructured data)
    - **BigText-KG:**

      A Hybrid Distributed **K**nowledge **G**raph Database



How to create the **BigText-KG** (Distributed **K**nowledge **G**raph Database)?

BigText-KG: Distributed Knowledge Graph Database



BigText-KG: Distributed Knowledge Graph Database

UNIVERSITÉ DU LUXEMBOURG

document title

sentence

entity

Thora Birch

From Wikipedia, the free encyclopedia

Thora Birch (born March 11, 1982)[2] is an American actress and producer. She made her film debut in *Purple People Eater* (1988), for which she won a Young Artist Award for "Best Young Actress Under Nine Years of Age". Birch rose to prominence as a child star with appearances in films including *All I Want for Christmas* (1991), *Patriot Games* (1992), *Hocus Pocus* (1993), *Monkey Trouble* (1994), *Now and Then* (1995), and *Alaska* (1996).

Her breakthrough role came in 1999 when she played Jane Burnham in the highly acclaimed film *American Beauty* for which she earned a BAFTA nomination for Best Supporting Actress. She then starred as Enid in the cult hit *Ghost World* (2001), earning a nomination for the Golden Globe for Best Actress. In 2003, Birch received an Emmy Award nomination for playing the title role in *Homeless to Harvard: The Liz Murray Story*. Her other films include *Dungeons & Dragons* (2000), *The Hole* (2001), *Silver City* (2004), *Dark Corners* (2006), *Winter of Frozen Dreams* (2009), and *Petunia* (2012).

American Beauty (1999 film)

From Wikipedia, the free encyclopedia

American Beauty is a 1999 American black comedy-drama film written by Alan Ball and directed by Sam Mendes. Kevin Spacey stars as Lester Burnham, an advertising executive who has a midlife crisis when he becomes infatuated with his teenage daughter's best friend, played by Mena Suvari. Annette Bening stars as Lester's materialistic wife, Carolyn, and Thora Birch plays their insecure daughter, Jane. Wes Bentley, Chris Cooper, and Allison Janney also feature. Academics have described the film as a satire of American middle class notions of beauty and personal satisfaction; further analysis has focused on the film's explorations of romantic and paternal love, sexuality, materialism, self-liberation, and redemption.

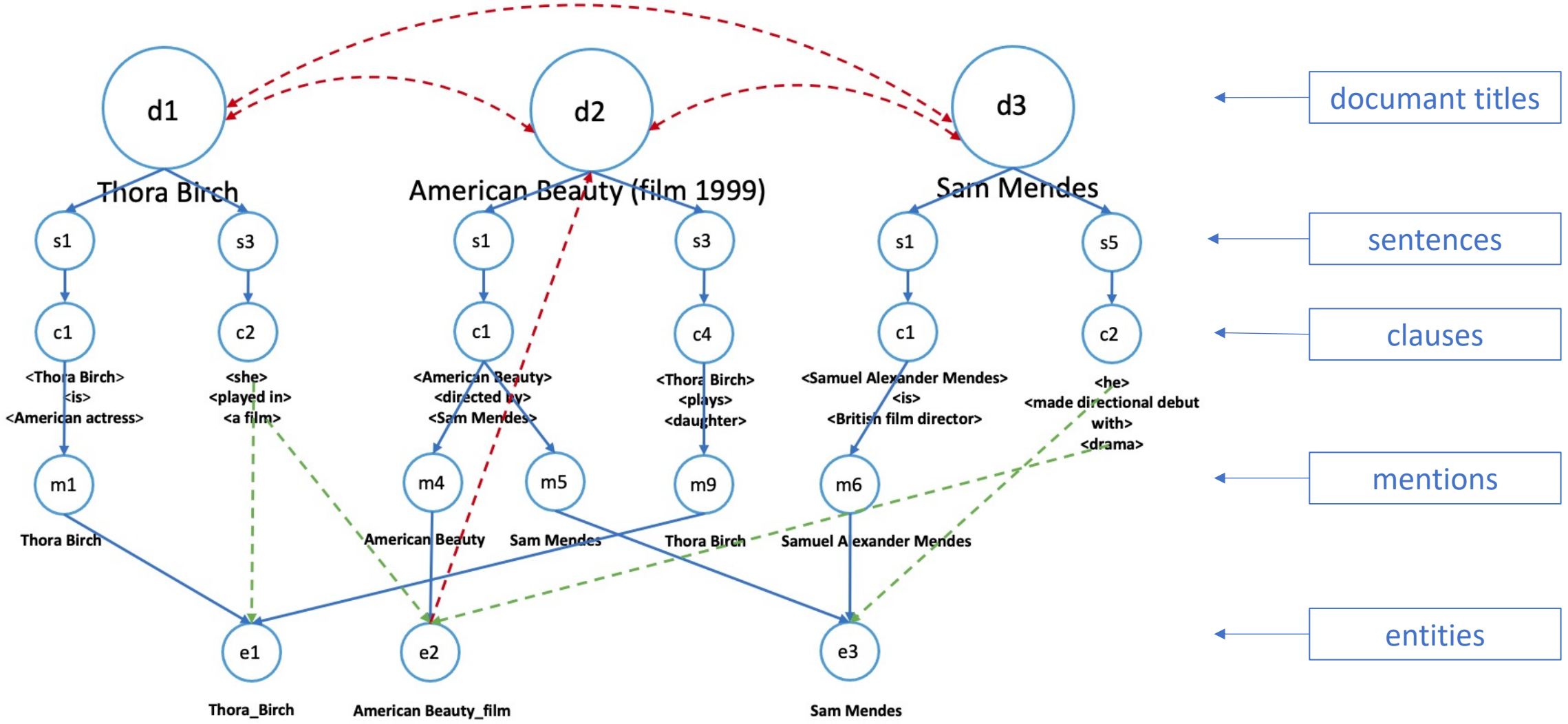
Sam Mendes

From Wikipedia, the free encyclopedia

Sir Samuel Alexander Mendes CBE (born 1 August 1965[1]) is a British film and stage director, producer, and screenwriter. In theatre, he is known for his dark re-inventions of the stage musicals *Cabaret* (1993), *Oliver!* (1994), *Company* (1995), and *Gypsy* (2003). He directed an original West End stage musical for the first time with *Charlie and the Chocolate Factory* (2013). For directing the play *The Ferryman*, Mendes was awarded the Tony Award for Best Direction of a Play in 2019.

In film, he made his directorial debut with the drama *American Beauty* (1999), which earned him the Academy Award and Golden Globe Award for Best Director. He has since directed the crime film *Road to Perdition* (2002), Jarhead (2005), the drama

BigText-KG: Distributed Knowledge Graph Database



BigText-KG: Distributed Knowledge Graph Database

- What is the size of the graph?
  - Graph Size:
    - **~570 million Nodes (Vertices)**:
      - Documents: 5.8 million
      - Sentences: 98 million
      - Clauses: 190 million
      - Mentions: 283 million
      - Entities: 2 million
    - **~780 million Edges**:
- How was it stored?
  - Location:
    - a single large Intel Xeon Platinum **server** with **2.4 GHz**, **192 virtual cores** and **1.2 TB of RAM**, holding the entire BigText-KG in main memory.
- What does it look like?
  - Dataframe example: pyspark with pair-rdd (key, value)

RDD: resilient distributed dataset;
Paired RDD: the RDD containing a key-value pair.

What does it look like? - Nodes
Dataframe example: pyspark with pair-rdd (key, value)

UNIVERSITÉ DU LUXEMBOURG

```
documentV\
.withColumnRenamed('attr1','documentTitle')\
.drop('attr2','attr3')\
.show(5, False)
```

1 document

```
+---------+--------+-------------------------+
|id       |nodeType|documentTitle            |
+---------+--------+-------------------------+
|379672629|Document|Lumileds                 |
|535344616|Document|Amanita luteofusca       |
|535345060|Document|Olive Hill, North Carolina|
|358932505|Document|Henriette Löfman         |
|363332769|Document|Christopher Curry (actor) |
+---------+--------+-------------------------+
only showing top 5 rows
```

Nodes

```
sentenceV\
.withColumnRenamed('attr1','serialNumber')\
.withColumnRenamed('attr2','sentence')\
.drop('attr1','attr2','attr3')\
.show(3, False)
```

2 sentences

```
+---------+--------+------------+----------------------------------------------------------------------------------------------------+
|id       |nodeType|serialNumber|sentence                                                                                            |
+---------+--------+------------+----------------------------------------------------------------------------------------------------+
|292357311|Sentence|45          |It soon became evident that no more would be ordered.                                               |
|292357755|Sentence|54          |This album includes a version of "Jerusalem" from the Black Sabbath album "Tyr" (1990).|
|292358199|Sentence|21          |"The London" again ceased publication with the issue dated 5 April 1879.                            |
+---------+--------+------------+----------------------------------------------------------------------------------------------------+
only showing top 3 rows
```

```
clauseV\
.withColumnRenamed('attr1','clause')\
.withColumnRenamed('attr2','clauseStructure')\
.drop('attr1','attr2','attr3')\
.show(5, False)
# S = Subject
# V = Verb
# O = Object
# A = Adverbial
# C = Complement
```

3 clauses

```
+---------+--------+-----------------------------------------------------------------------------------------------------------------------+---------------+
|id       |nodeType|clause                                                                                                                 |clauseStructure|
+---------+--------+-----------------------------------------------------------------------------------------------------------------------+---------------+
|274170134|Clause  |("The term tickle torture", "can apply", "to many different situations")                                               |SVA            |
|274170578|Clause  |("The attraction", "houses", "the world's largest private aircraft collection", "on display")                          |SVO            |
|274171022|Clause  |("The Continuo World Champions from 1997 onwards", "are")                                                              |SV             |
|274171466|Clause  |("Velvet McIntyre born November 24 1962", "is", "an wrestler", "Irish-Canadian", "retired", "professional")            |SVC            |
|274171910|Clause  |("This relaxation in the regulations", "has made", "mid and high power rocketry much more accessible", "in the UK")|SVO            |
+---------+--------+-----------------------------------------------------------------------------------------------------------------------+---------------+
only showing top 5 rows
```

```
entityV\
.withColumnRenamed('attr1','entity')\
.withColumn('entity', f_spark.split('entity', ' ').getItem(0))\
.drop('attr1','attr2','attr3')\
.show(5, False)
```

5 entities

```
+---------+--------+-----------------------------------+
|id       |nodeType|entity                             |
+---------+--------+-----------------------------------+
|578078738|Entity  |Żakowice,_Warmian-Masurian_Voivodeship|
|578079182|Entity  |Chemical_Markup_Language           |
|578079626|Entity  |Boone_Grove,_Indiana               |
|578080070|Entity  |Givenchy                           |
|578080514|Entity  |Sweet_Fever                        |
+---------+--------+-----------------------------------+
only showing top 5 rows
```

```
mentionV\
.withColumnRenamed('attr1','mention')\
.withColumnRenamed('attr2','mentionType')\
.drop('attr1','attr2','attr3')\
.show(5, False)
```

4 mentions

```
+---------+--------+-------------------+-----------+
|id       |nodeType|mention            |mentionType|
+---------+--------+-------------------+-----------+
|314295501|Mention |Freddy             |PERSON     |
|314295945|Mention |Latin              |LOCATION   |
|314296389|Mention |New York           |LOCATION   |
|314296833|Mention |Eugene Joseph Carey|PERSON     |
|314297277|Mention |Robert John Bardo  |PERSON     |
+---------+--------+-------------------+-----------+
only showing top 5 rows
```

What does it look like? - Nodes
Dataframe example: pyspark with pair-rdd (key, value)

UNIVERSITÉ DU LUXEMBOURG

Sentence Nodes

```
sentenceV\
.withColumnRenamed('attr1','serialNumber')\
.withColumnRenamed('attr2','sentence')\
.drop('attr1','attr2','attr3')\
.show(3, False)
```

```
+---------+--------+-----------+-----------------------------------------------------------------------------+
|id       |nodeType|serialNumber|sentence                                                                     |
+---------+--------+-----------+-----------------------------------------------------------------------------+
|292357311|Sentence|45         |It soon became evident that no more would be ordered.                        |
|292357755|Sentence|54         |This album includes a version of "Jerusalem" from the Black Sabbath album "Tyr" (1990).|
|292358199|Sentence|21         |"The London" again ceased publication with the issue dated 5 April 1879.     |
+---------+--------+-----------+-----------------------------------------------------------------------------+
only showing top 3 rows
```

What does it look like? - Edges
Dataframe example: pyspark with pair-rdd (key, value)

Edges

document - sentence

```
edges.filter(f_spark.col('label')=='contains the sentence').show(3, False)
```

```
+---------+---------+--------------------+
|src      |dst      |label               |
+---------+---------+--------------------+
|557332961|557840303|contains the sentence|
|557332961|557840304|contains the sentence|
|248782795|250469936|contains the sentence|
+---------+---------+--------------------+
only showing top 3 rows
```

sentence - clause

```
edges.filter(f_spark.col('label')=='contains the clause').show(3, False)
```

```
+---------+---------+------------------+
|src      |dst      |label             |
+---------+---------+------------------+
|455730908|454209661|contains the clause|
|455730908|454209660|contains the clause|
|257211306|258422514|contains the clause|
+---------+---------+------------------+
only showing top 3 rows
```

clause - mention

```
edges.filter(f_spark.col('label')=='contains the mention').show(3, False)
```

```
+---------+---------+------------------+
|src      |dst      |label             |
+---------+---------+------------------+
|571977434|572691721|contains the mention|
|69594473 |70235232 |contains the mention|
|69594473 |70235233 |contains the mention|
+---------+---------+------------------+
only showing top 3 rows
```

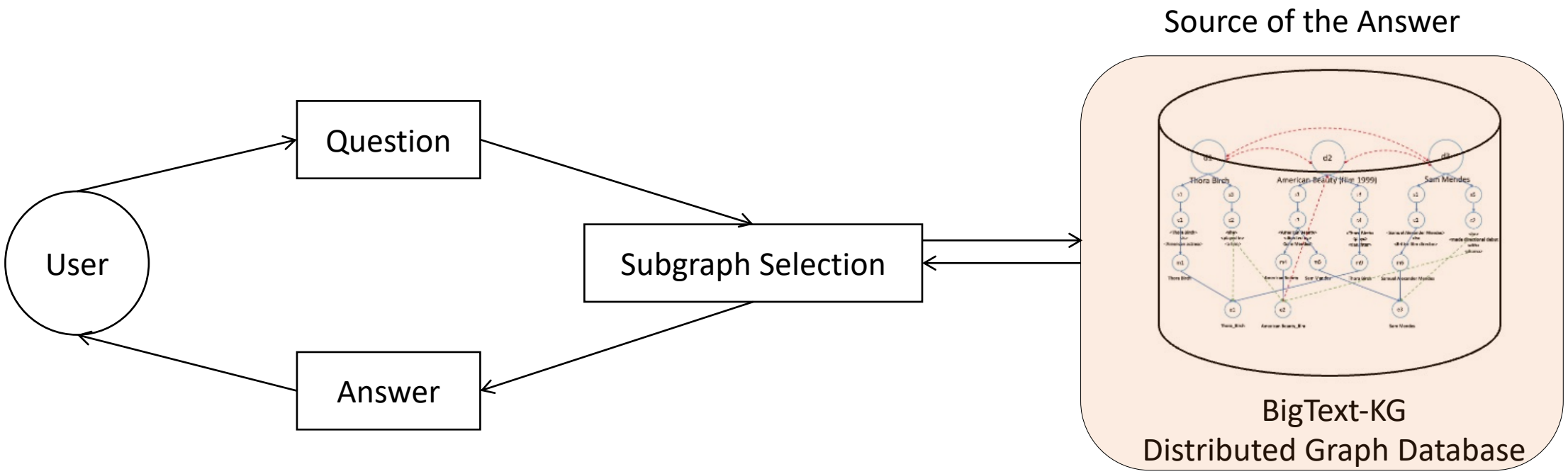
mention - entity

```
edges.filter(f_spark.col('label')=='is disambiguated as').show(3, False)
```

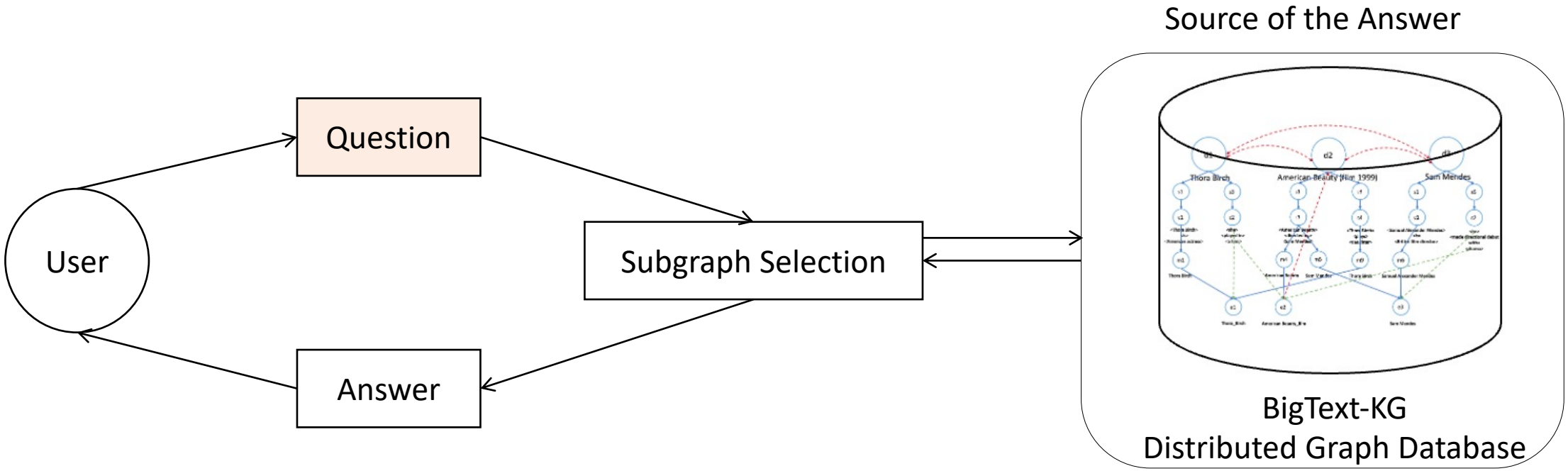
```
+--------+---------+----------------+
|src     |dst      |label           |
+--------+---------+----------------+
|11293332|578838527|is disambiguated as|
|11293335|578838527|is disambiguated as|
|11293336|578838527|is disambiguated as|
+--------+---------+----------------+
only showing top 3 rows
```

How to build **BigText-QA** from **BigText-KG** (Distributed **K**nowledge **G**raph Database)?

BigText-QA: Pipeline



BigText-QA: Pipeline



Question assumptions:

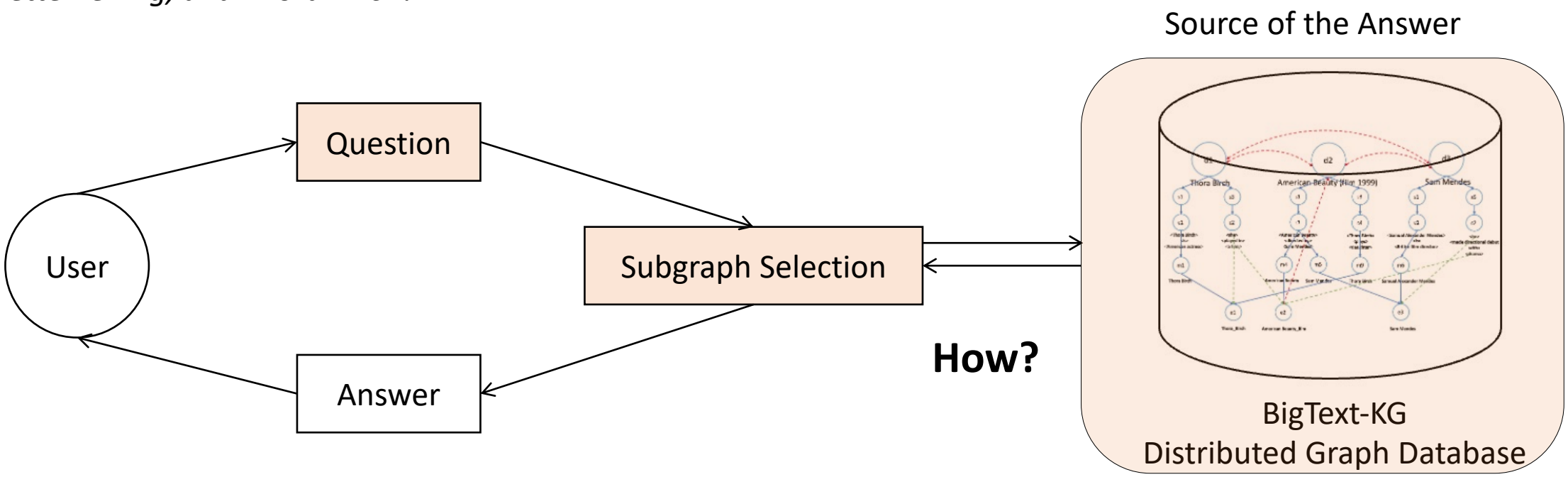
- Complex question with **multiple named entities**,
- **multiple relationships** connecting these entities,
- and a **single typed target entity**

Example question (from CQ-W dataset):

- ***Which British stage director is best known for his feature-film directing debut, which starred Kevin Spacey, Annette Bening, and Thora Birch?***

BigText-QA: Pipeline

Question: Which British stage director is best known for his feature-film directing debut, which starred Kevin Spacey, Annette Bening, and Thora Birch?



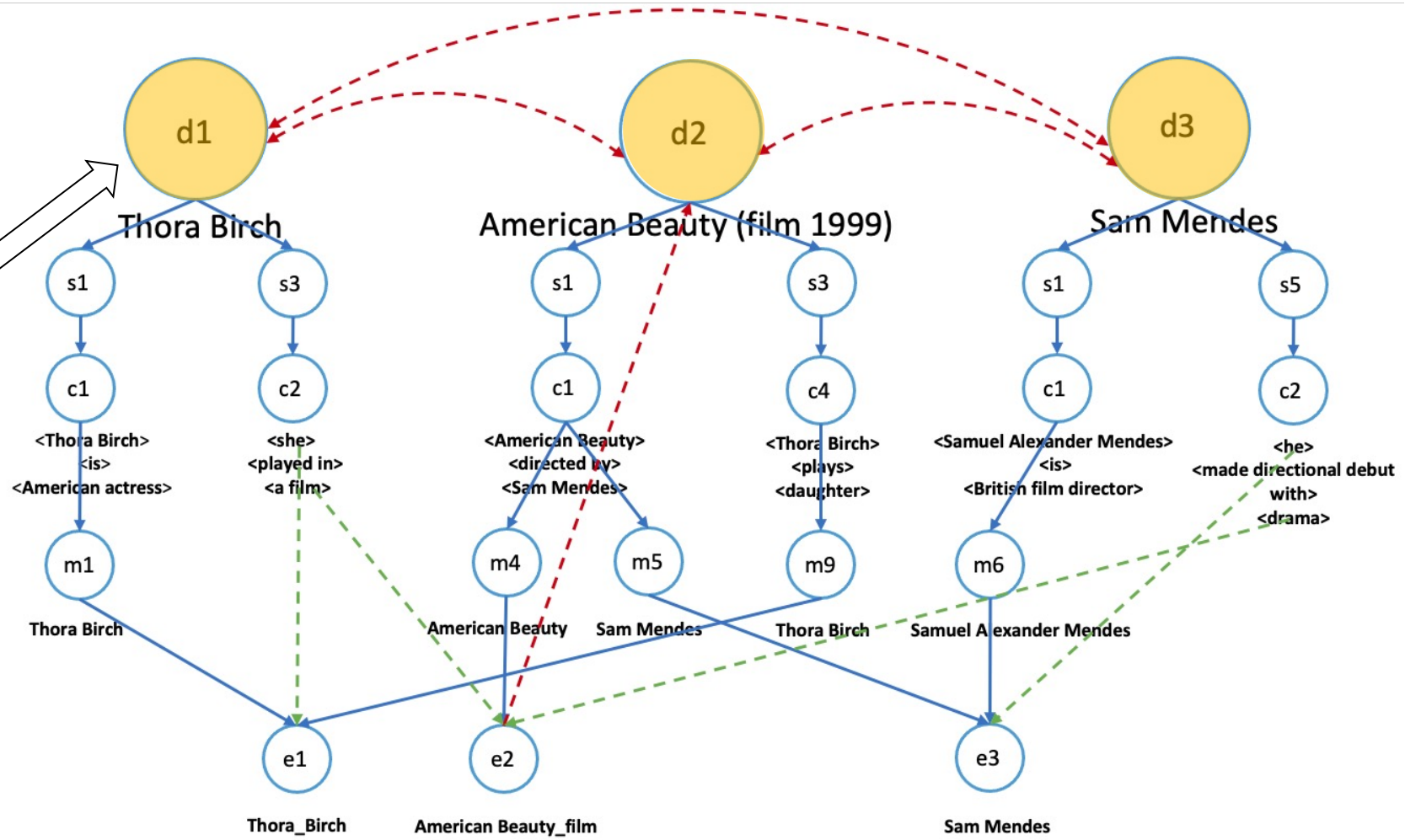
BigText-QA: Subgraph Selection

Question:
Which British stage director is best known for his feature-film directing debut, which starred Kevin Spacey, Annette Bening, and Thora Birch?

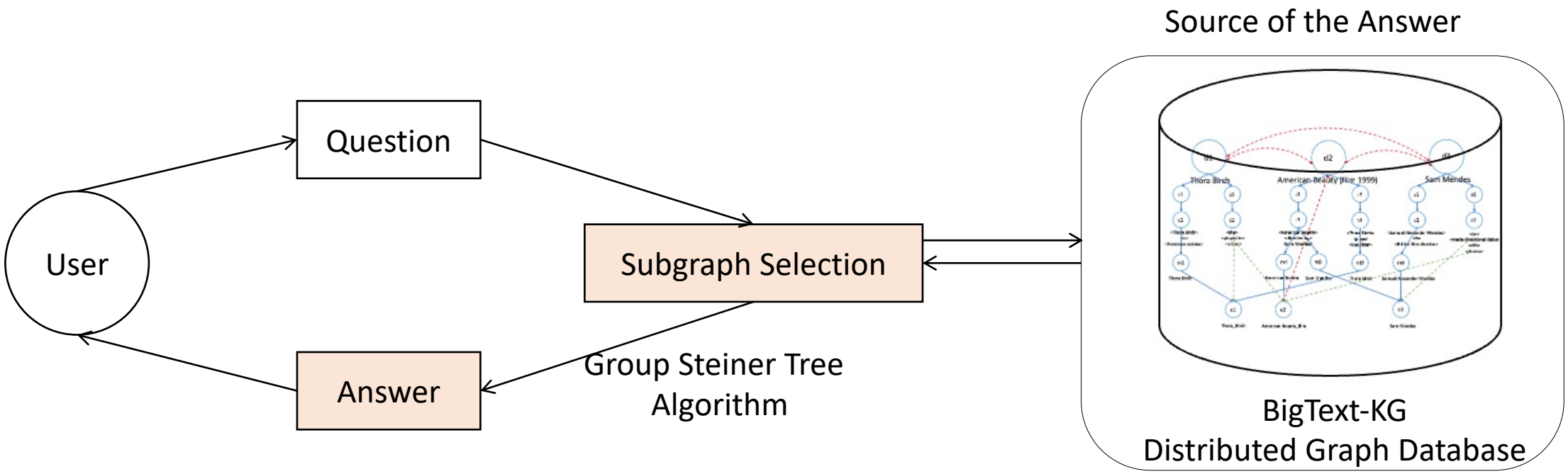
Lucene

top related documents

Documents' titles



BigText-QA: Pipeline



BigText-QA: Results

Dataset	System	Cosine	#Vertices	#Edges (10^5)	MRR	P@1	Hit@5
CQ-W	BigText-QA	0.250	1,276	7.234	0.387	0.324	0.441
		0.375	1,276	7.234	0.387	0.324	0.441
		0.500	**1,268**	6.727	**0.398**	**0.342**	**0.423**
		0.600	**579**	0.510	**0.264**	**0.198**	**0.297**
		0.750	210	0.030	0.140	0.081	0.189
	QUEST	0.500	2,385	13.580	0.464	0.423	0.495
		0.600	**1,267**	0.609	0.329	0.279	0.369
		0.750	**642**	0.032	0.181	0.099	0.279
	DrQA	-	-	-	0.120	0.171	0.315
Trivia-QA	BigText-QA	0.375	840	2.073	0.412	0.342	0.494
		0.500	**838**	1.968	**0.412**	**0.342**	**0.468**
		0.600	**365**	0.163	**0.258**	**0.190**	**0.316**
		0.750	121	0.007	0.130	0.063	0.190
	QUEST	0.500	1,710	4.025	0.425	0.380	0.468
		0.600	**968**	0.241	0.285	0.215	0.329
		0.750	**490**	0.025	0.198	0.139	0.241

graph-based QA

neural-network-based QA

Table 1. Comparison between BigText-QA, QUEST and DrQA on the CQ-W and TriviaQA datasets (a proof-of-concept for BigText Graph Database).

References

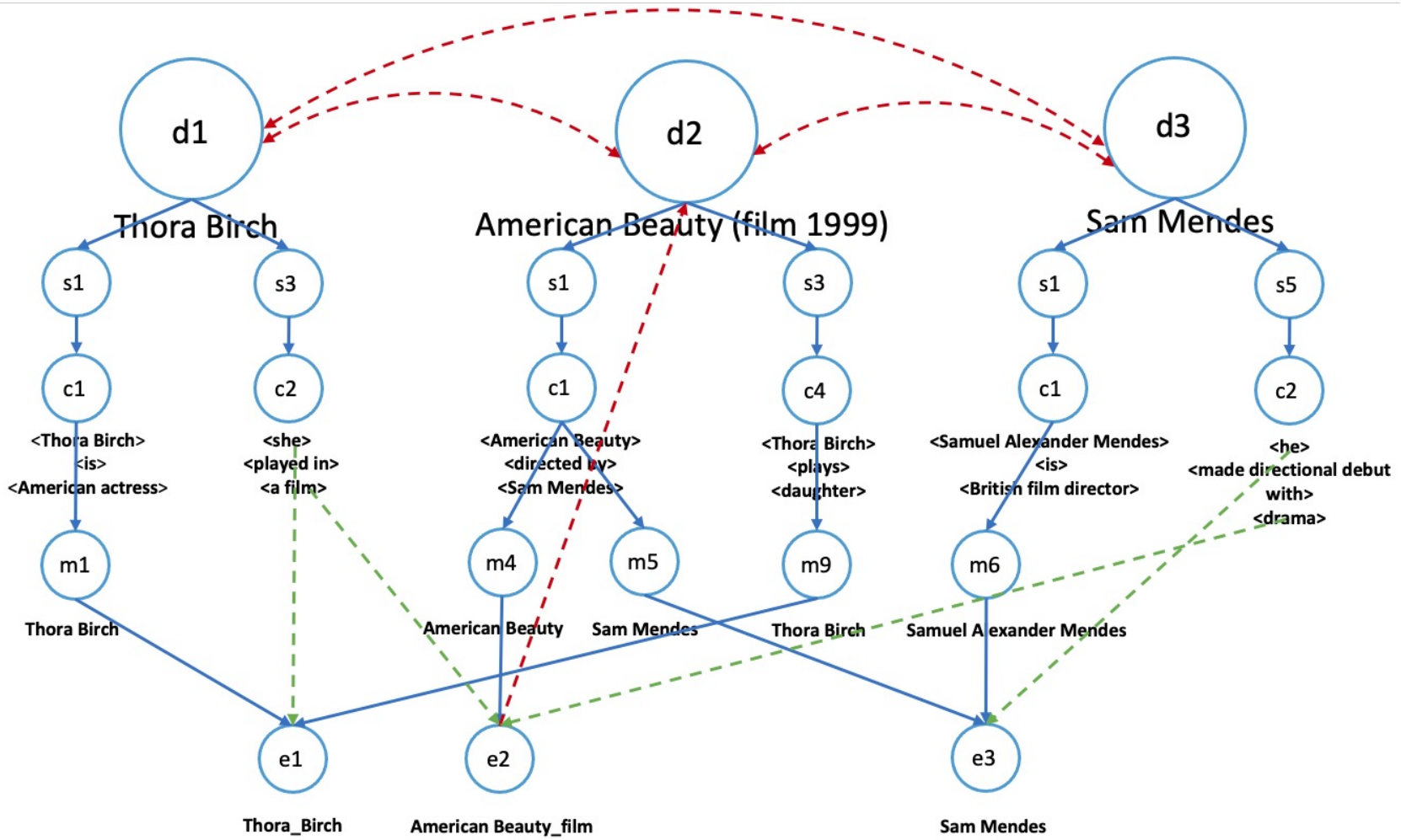
[1] Lu, Xiaolu, et al. "Answering complex questions by joining multi-document evidence with quasi knowledge graphs." Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019.

[2] Chen, Danqi, et al. "Reading wikipedia to answer open-domain questions." arXiv preprint arXiv:1704.00051 (2017).

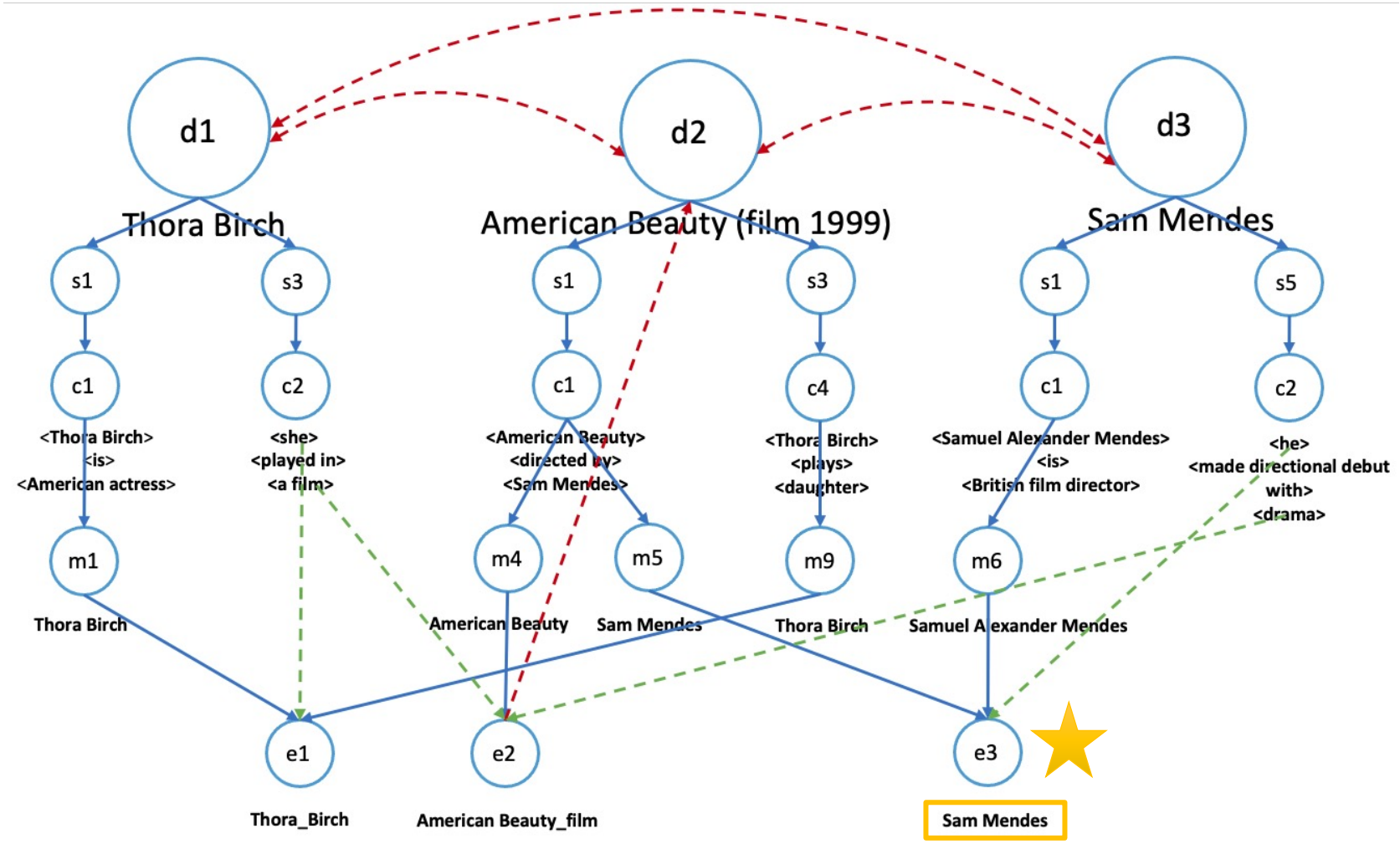
[3] Joshi, Mandar, et al. "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension." arXiv preprint arXiv:1705.03551 (2017).

Acknowledgement

Question:
Which British stage director is best known for his feature-film directing debut, which starred Kevin Spacey, Annette Bening, and Thora Birch?



Question:
Which British stage director is best known for his feature-film directing debut, which starred Kevin Spacey, Annette Bening, and Thora Birch?




Thank You!
Q & A